

(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-304331

(P2002-304331A)

(43) 公開日 平成14年10月18日 (2002. 10. 18)

(51) Int. Cl. <sup>7</sup>	識別記号	F I	テーマコード(参考)
G 0 6 F 13/00	3 0 1	G 0 6 F 13/00	3 0 1 Q 5 B 0 1 4
3/06	3 0 5	3/06	3 0 5 K 5 B 0 6 5
13/10	3 4 0	13/10	3 4 0 A 5 B 0 8 3

審査請求 有 請求項の数 8 O L (全 7 頁)

(21) 出願番号 特願2001-106927(P2001-106927)

(22) 出願日 平成13年4月5日(2001.4.5)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 三木 健一

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100079164

弁理士 高橋 勇

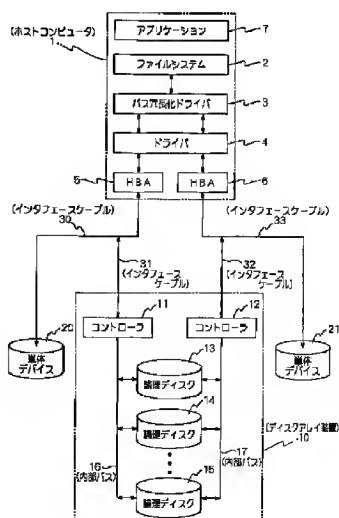
最終頁に続く

## (54) 【発明の名称】 冗長バス制御装置及び方法

## (57) 【要約】

【課題】 リザーブされた論理ディスクに対しても、代替バスを用いてアクセスする。

【解決手段】 バス冗長化ドライバ3は、論理ディスク13が他のイニシエータ5によってリザーブされているためにリトライ処理が異常終了した場合は、代替バスのイニシエータ6から強制リザーブコマンドを発行する。コントローラ12は、強制リザーブコマンドが発行されると、代替バスのイニシエータ6によるリザーブ状態に論理ディスク13を設定し直す。



## 【特許請求の範囲】

【請求項1】 1／0アクセスバスに障害が発生した場合に、イニシエータを含む代替バスを用いて論理ディスクに対してリトライ処理を行う冗長バス制御装置において、

前記論理ディスクが他のイニシエータによってリザーブされているために前記リトライ処理が異常終了した場合は、前記代替バスのイニシエータによるリザーブ状態に前記論理ディスクを設定し直す冗長バス制御手段、を備えたことを特徴とする冗長バス制御装置。

【請求項2】 前記冗長バス制御手段は、前記論理ディスクが他のイニシエータによってリザーブされているために前記リトライ処理が異常終了した場合は、前記代替バスのイニシエータから強制リザーブコマンドを発行する第一の制御手段と、前記強制リザーブコマンドが発行されると、前記代替バスのイニシエータによるリザーブ状態に前記論理ディスクを設定し直す第二の制御手段と、を備えた請求項1記載の冗長バス制御装置。

【請求項3】 前記論理ディスクが複数集まってディスクアレイ装置を構成している、

請求項1又は2記載の冗長バス制御装置。  
【請求項4】 前記イニシエータがHBA（Host Bus Adapter）である、

請求項1、2又は3記載の冗長バス制御装置。  
【請求項5】 1／0アクセスバスに障害が発生した場合に、イニシエータを含む代替バスを用いて論理ディスクに対してリトライ処理を行う冗長バス制御方法において、

前記論理ディスクが他のイニシエータによってリザーブされているために前記リトライ処理が異常終了した場合は、前記代替バスのイニシエータによるリザーブ状態に前記論理ディスクを設定し直す冗長バス制御ステップ、を備えたことを特徴とする冗長バス制御方法。

【請求項6】 前記冗長バス制御ステップは、前記論理ディスクが他のイニシエータによってリザーブされているために前記リトライ処理が異常終了した場合は、前記代替バスのイニシエータから強制リザーブコマンドを発行する第一の制御ステップと、前記強制リザーブコマンドが発行されると、前記代替バスのイニシエータによるリザーブ状態に前記論理ディスクを設定し直す第二の制御ステップと、を備えた請求項5記載の冗長バス制御方法。

【請求項7】 前記論理ディスクが複数集まってディスクアレイ装置を構成している、

請求項5又は6記載の冗長バス制御方法。  
【請求項8】 前記イニシエータがHBA（Host Bus Adapter）である、

請求項5、6又は7記載の冗長バス制御方法。

【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、1／0アクセスバスに障害が発生した場合に、イニシエータを含む代替バスを用いて論理ディスクに対してリトライ処理を行う、冗長バス制御装置及び方法に関する。

## 【0002】

【従来の技術】バス冗長化ドライバ3'は、1／0アクセスバスの構成要素すなわちHBA（Host Bus Adapter）5、6、インタフェースケーブル31、32、コントローラ11'、12'等における障害（故障）が発生した場合においても、ホストコンピュータ1上で動作するアプリケーション7の運用に影響を与えないために、代替バスにてリトライ処理を行う手段を提供するものである。すなわち、冗長化されたディスクアレイ装置10とホストコンピュータ1上で動作するソフトウェア（バス冗長化ドライバ3'）とにより、論理ディスク13～15に対する1／0バスの冗長性を実現することができる。

【0003】また、ホストコンピュータ1上で動作するアプリケーション7が使用する任意の論理ディスク13～15に対し、図示しない他のアプリケーション又は他のホストコンピュータからアクセスができないように占有（リザーブ）する場合がある。一般的に、この占有状態は、以前占有を行ったイニシエータ（ここではHBAが該当）からの解除コマンドによって解除されるか、又はHW（ハードウェア）によるリセットによって強制的に解除される。

## 【0004】

【発明が解決しようとする課題】ここで例えば、ホストコンピュータ1上で動作するアプリケーション7が、HBA5、インタフェースケーブル31及びコントローラ11'経由で論理ディスク13を占有しつつ使用している場合を考えると、次のような問題点が発生する。

【0005】第一の問題点は、次のような場合、バス冗長化ドライバ3'の機能が意味をなさなくなることである。バス冗長化ドライバ3'は、ホストコンピュータ1上で動作するアプリケーション7が論理ディスク13に対する1／0要求を行った実行結果をドライバ4から受け取り、正常終了又は異常終了の判断を行う。このとき、異常終了と判断し、更にその原因がバスの構成要素（HBA5、インタフェースケーブル31又はコントローラ11'）における障害（故障）であると判断したとする。その結果、代替バスを用いて、すなわち論理ディスク13に対しHBA6、インタフェースケーブル32及びコントローラ12'経由で、異常終了した1／0のリトライ処理を試みる。しかし、論理ディスク13は他のイニシエータ（HBA5）により占有状態にあるため、読み出し／書き込み等ができなく、結果としてアプリケーション7の1／0要求は異常終了することとなる。したがって、バス冗長化ドライバ3'は所期の目的

を果たすことができない。

【0006】第二の問題点は、HWにより強制的にリセットを行い占有状態を解除した場合、コントローラ11'、12'が初期化されることによりホストコンピュータ1側から受け取った処理中の一つ又は複数のI/O要求がすべて破棄されてしまうため、ホストコンピュータ1側で余計なリトライ処理が発生することである。また、バス（ホストインタフェース）をリセットした場合は、バスに接続されている全てのデバイス（単体デバイス20、21）も初期化されることとなるため、ホストコンピュータ1側に更に多くの負担（リトライ処理）をかけることである。

【0007】また、論理ディスク13～15上のデータ整合性の保護を目的として、ホストコンピュータ1が任意の論理ディスク13～15を占有している場合を考えると、次のような問題点が発生する。

【0008】第三の問題点は、あるホストコンピュータから実施した論理ディスクに対する占有状態を一時的に解除状態にすることは、まったく意図しないホストコンピュータからのアクセスを許すこととなるので、論理ディスク内のデータを破壊する恐れがあることである。

【0009】

【発明の目的】そこで、本発明の目的は、既にリザーブされている論理ディスクに対しても、代替バスを用いてアクセスできるようにした、冗長バス制御装置及び方法を提供することにある。

【0010】

【課題を解決するための手段】本発明に係る冗長バス制御装置（又は冗長バス制御方法）は、I/Oアクセスバスに障害が発生した場合に、イニシエータを含む代替バスを用いて論理ディスクに対してリトライ処理を行うものであり、論理ディスクが他のイニシエータによってリザーブされているためにリトライ処理が異常終了した場合は、代替バスのイニシエータによるリザーブ状態に論理ディスクを設定し直す冗長バス制御手段（又は冗長バス制御ステップ）を備えている（請求項1、5）。

【0011】論理ディスクに対して代替バスを用いてリトライ処理を実行したところ、その論理ディスクが既に他のイニシエータによってリザーブされていたため、リトライ処理が異常終了したとする。すると、冗長バス制御手段（又は冗長バス制御ステップ）は、その論理ディスクを、代替バスのイニシエータによるリザーブ状態に設定し直す。これにより、その論理ディスクに対して、代替バスを用いてアクセスできるようになる。

【0012】また、冗長バス制御手段（又は冗長バス制御ステップ）は、論理ディスクが他のイニシエータによってリザーブされているためにリトライ処理が異常終了した場合は、代替バスのイニシエータから強制リザーブコマンドを発行する第一の制御手段（又は第一の制御ステップ）と、強制リザーブコマンドが発行されると、代

替バスのイニシエータによるリザーブ状態に論理ディスクを設定し直す第二の制御手段（又は第二の制御ステップ）とを備えた、としてもよい（請求項2、6）。

【0013】更に、論理ディスクが複数集まってディスクアレイ装置を構成している（請求項3、7）、又はイニシエータがHBAである（請求項4、8）、としてもよい。

【0014】

【発明の実施の形態】以下、本発明に係る冗長バス制御装置の実施形態を、図面にに基づき説明する。なお、本発明に係る冗長バス制御方法は、本発明に係る冗長バス制御装置で使用されるものである。したがって、本実施形態を説明することにより、本発明に係る冗長バス制御方法の一実施形態も説明したことにする。

【0015】図1は、本発明に係る冗長バス制御装置の一実施形態を示すブロック図である。以下、この図面にに基づき説明する。なお、特許請求の範囲における第一の手段及び第二の手段は、それぞれ本実施形態におけるバス冗長化ドライバ及びコントローラに相当する。

【0016】ホストコンピュータ1のHBA5、6は、それぞれインタフェースケーブル31、32を介してディスクアレイ装置10のコントローラ11、12に接続されている。ホストコンピュータ1は、ディスクアレイ装置10内の論理ディスク13～15に対するI/Oを実行する。ドライバ4は、HBA5、6を制御しI/O処理を行う。

【0017】バス冗長化ドライバ3は、次のような動作をする。ファイルシステム2から受け取ったI/Oを、ドライバ4に引き渡す。ディスクアレイ装置10内の論理ディスク13～15のいずれかに対するI/Oの実行結果を、HBA5、6を介してドライバ4から受け取り、正常終了又は異常終了の判断を行う。異常終了の原因がバスの構成要素（HBA5、6、インタフェースケーブル31、32、コントローラ11、12等）における障害（故障）と判断した場合は、代替バスを用いて、異常終了したI/Oのリトライ処理を行う。

【0018】ディスクアレイ装置10のコントローラ11、12は、それぞれ内部バス16、17を介して各論理ディスク13～15に接続されている。コントローラ11、12のいずれからでも、各論理ディスク13～15に対してアクセスすることができる。

【0019】例えば、バス冗長化ドライバ3は、論理ディスク13が他のイニシエータ5によってリザーブされているためにリトライ処理が異常終了した場合は、代替バスのイニシエータ6から強制リザーブコマンドを発行する。コントローラ12は、強制リザーブコマンドが発行されると、代替バスのイニシエータ6によるリザーブ状態に論理ディスク13を設定し直す。

【0020】図2は、ホストコンピュータ1から受け取ったI/Oコマンド処理過程の一部を示したフローであ

り、これらの処理はディスクアレイ装置10のコントローラ11、12により実行される。図3は、ドライバ4から受け取ったI/Oの実行結果に対し、正常終了又は異常終了の判断を行い、異常終了の原因がバスの構成要素(HBA5、6、インタフェースケーブル31、32、コントローラ11、12等)における障害(故障)と判断した場合は、代替バスを用いて、異常終了したI/Oのリトライ処理をする過程の一部を示したフローであり、これらの処理はバス冗長化ドライバ3により実施される。以下、図1乃至図3に基づき、本実施形態の冗長バス制御装置の動作を説明する。

【0021】ホストコンピュータ1上で動作するアプリケーション7によってディスクアレイ装置10に書き込まれるデータ(ライトI/O)は、アプリケーション7、ファイルシステム2、バス冗長化ドライバ3、ドライバ4、HBA5、インタフェースケーブル31を介してコントローラ11に至り、指定された論理ディスク13〜15のいずれかに書き込まれる。

【0022】ホストコンピュータ1上で動作するアプリケーション7によってディスクアレイ装置10から読み出されるデータ(リードI/O)は、指定された論理ディスク13〜15のいずれかより、コントローラ11、インタフェースケーブル31を介してHBA5に至り、ドライバ4、バス冗長化ドライバ3、ファイルシステム2を経てアプリケーション7に至る。

【0023】また、ホストコンピュータ1による各I/Oの実行結果については、HBA5、ドライバ4、ファイルシステム2及びアプリケーション7の各レイヤによって判断が行われ、必要に応じて何らかの処置が行われるのが一般的である。ここで、バス冗長化ドライバ3は、ドライバ4から受け取ったI/Oの実行結果について正常終了又は異常終了の判断を行い、異常終了の原因がバスの構成要素(HBA、インタフェースケーブル、コントローラ等)における障害(故障)と判断した場合は、代替バスを用いて、異常終了したI/Oのリトライ処理を行う。

【0024】次に、アプリケーション7が論理ディスク13をリザーブした状態について説明する。

【0025】リザーブコマンドとは、特定のイニシエータ(HBA)が指定した論理ディスクを排他的に占有するためのコマンドである。例えば、HBA5からコントローラ11経由で論理ディスク13を指定してリザーブが行われている場合、HBA5からコントローラ11経由による論理ディスク13に対するI/O(リード又はライト)は、障害などの問題のない限り正常終了する。しかし、HBA6からコントローラ12経由で論理ディスク13に対して発行されたI/O(リード又はライト)は、「Reservation Conflict」というエラーメッセージと共に異常終了する。

【0026】これらのことからわかるように、リザーブ

コマンドによって占有された論理ディスク13に対するI/Oに対し、バス冗長化ドライバ3は、その機能うまく働かせることができない。

【0027】この問題を解決するための方法を図2及び図3のフローを用いて説明する。

【0028】図2は、ディスクアレイ装置10のコントローラ11、12により実施され、ホストコンピュータ1から受け取ったI/Oコマンド処理過程の一部を示したフローである。本問題を解決するために、ディスクアレイ装置10にて強制リザーブ手段を設ける。

【0029】まず、ホストコンピュータ1から受け取ったコマンドが、まず正しく実行できるものであるか否かを判断する(S101)。正しく実行することができないコマンドであれば、異常終了処理へ移行する(S111)。正しく実行可能なコマンドであれば、それが既存のコマンドであるか、又は新たに設けた強制リザーブコマンドであるかを判断する(S102)。強制リザーブコマンドであればフラグをクリアし(S103)、そうでなければ既存コマンドデコード処理へ移行する(S112)。

【0030】続いて、指定された論理ディスクが既にリザーブされているか否かを判断する(S104)。リザーブされていない場合は、強制リザーブコマンドを発行したイニシエータによるリザーブ状態に設定する(S107)。既にリザーブされている場合は、そのリザーブが強制リザーブコマンドを発行したイニシエータとは別のイニシエータによるリザーブであるかを判断する(S105)。強制リザーブコマンドを発行したイニシエータからのリザーブであれば、終了処理へ移行する(S113)。強制リザーブコマンドを発行したイニシエータとは別のイニシエータによるリザーブであった場合は、フラグをセットし(S106)、強制リザーブコマンドを発行したイニシエータによるリザーブ状態に設定する(S107)。

【0031】続いて、フラグの確認を行ない(S108)、フラグがセットされていない場合は、終了処理へ移行する(S110)。フラグがセットされている場合は、他のイニシエータからリザーブされているため、そのリザーブ状態を解除するとともに使用したフラグをクリアし(S109)、終了処理へ移行する(S110)。

【0032】図3は、バス冗長化ドライバ3により実施され、代替バスによるリトライ処理過程の一部を示したフローである。

【0033】まず、バス冗長化ドライバ3は、ディスクアレイ装置10内の論理ディスク1〜15に対するI/Oの実行結果をドライバ4から受け取り、正常終了又は異常終了の判断を行う。そして、異常終了の原因がバスの構成要素(HBA、インタフェースケーブル、コントローラ等)における障害(故障)と判断した場合は、代

替バスを用いて、異常終了したI/Oのリトライ処理を行う(Ｓ201)。続いて、リトライ結果を判断し(Ｓ202)、異常がなければ正常終了処理(Ｓ209)へ移行する。

【0034】異常があれば、異常の原因が他のイニシエータによってリザーブされた論理ディスクによるものであるかを判断する(Ｓ203)。他に起因する異常の場合は、異常終了処理へ移行する(Ｓ210)。一方、異常の原因が他のイニシエータによってリザーブされた論理ディスクによるものである場合は、同一バスに

対して強制リザーブコマンドを実施する(Ｓ204)。【0035】続いて、その結果の判断を行なう(Ｓ205)。強制リザーブコマンドが正常終了でなければ、異常終了処理へ移行する(Ｓ211)。強制リザーブコマンドが正常終了であれば、論理ディスクに対する占有が本イニシエータに変更されているため、ステップ202にて異常終了となったI/Oを同一バスにて再度リトライを実施する(Ｓ206)。続いて、ステップ206の結果を判断し(Ｓ207)、リトライが成功した場合は正常終了処理へ移行し(Ｓ208)、失敗であれば異常終了処理へ移行する(Ｓ212)。

【0036】以上により、リザーブされた論理ディスクに対するI/Oバス冗長化システムを実現する。

【0037】なお、本発明は、言うまでもなく、上記実施形態に限定されるものではない。例えば、次のような実施の形態を探り得る。

【0038】HBA 5、6の数は、OSの種類、ドライバ4、又はホストコンピュータ1等のHWによって制限され、バス冗長化ドライバ3に対して制限はない。コントローラ11、12の数、及びインタフェースケーブル31、32を介して各コントローラ11、12とホストコンピュータ1等を接続するためのポート数に制限はない。HBA 5、6とコントローラ11、12とを直接インタフェースケーブル31、32で接続したものを例示したが、これらの途中にハブ又はスイッチ等が介在してもかまわない。

【0039】ホストコンピュータ1に接続されるディスクアレイ装置10の数に制限はない。ディスクアレイ装置10に接続されるホストコンピュータ1の数に制限はない。ディスクアレイ装置10内の論理ディスク13～14の数に制限はない。本発明はディスクアレイ装置10のみに限定されるものではない。

【0040】

【発明の効果】本発明に係る冗長バス制御装置及び方法によれば、論理ディスクが他のイニシエータによって既にリザーブされているためにリトライ処理が異常終了した場合に、代替バスのイニシエータによるリザーブ状態に当該論理ディスクを設定し直すことにより、当該論理ディスクに対して代替バスを用いてアクセスすることができる。換言すると、バスの構成要素(HBA、インタフェースケーブル、コントローラ等)における障害(故障)時に、論理ディスクを占有して使用するアプリケーションが存在する場合も、アプリケーションの運用に影響を与えることのないよう代替バスでのリトライが可能となる。

【0041】また、HWによるリセットによって強制的に占有の解除を行わないため、ホストコンピュータ側への余計なリトライ処理の発生を抑制できる。

【0042】更に、論理ディスクに対する占有状態を一時的にでも解除状態にしないことにより、全く意図しないホストコンピュータからのアクセスを許す契機を与えないので、論理ディスク内のデータを破壊する危険を回避できる。

#### 【図面の簡単な説明】

【図1】本発明に係る冗長バス制御装置の一実施形態を示すブロック図である。

【図2】図1の冗長バス制御装置におけるコントローラの動作の一例を示すフローチャートである。

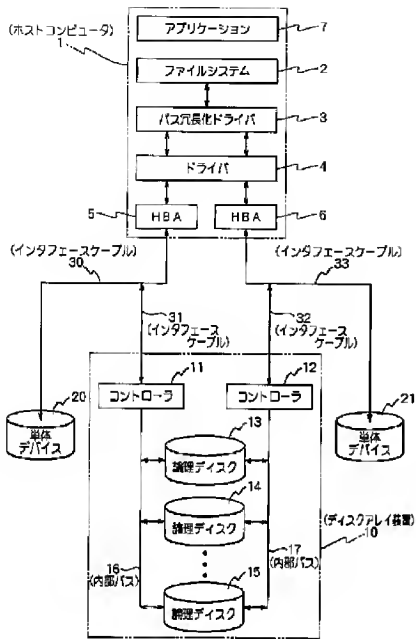
【図3】図1の冗長バス制御装置におけるバス冗長化ドライバの動作の一例を示すフローチャートである。

【図4】従来の冗長バス制御装置を示すブロック図である。

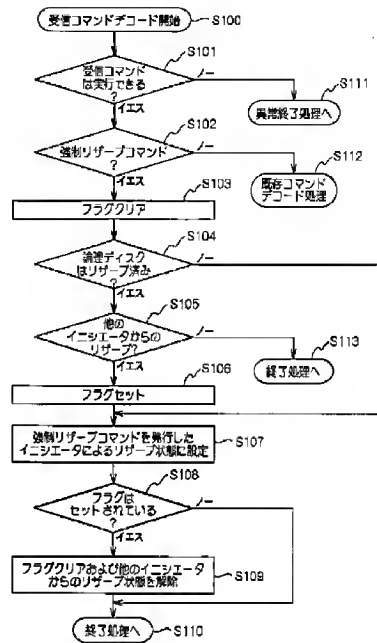
#### 【符号の説明】

- 1 ホストコンピュータ
- 2 ファイルシステム
- 3 バス冗長化ドライバ
- 4 ドライバ
- 5、6 HBA
- 7 アプリケーション
- 30～33 インタフェースケーブル
- 10 ディスクアレイ装置
- 11、12 コントローラ
- 13～15 論理ディスク
- 16、17 内部バス
- 20、21 単体ドライバ

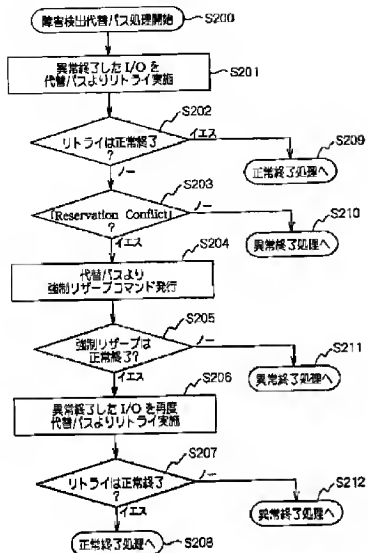
【図1】



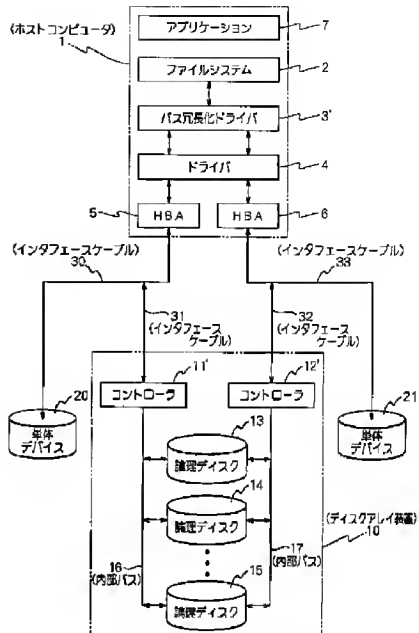
【図2】



【図3】



【図4】



フロントページの続き

Fターム(参考) 5B014 EB04 FA04 GA15 GD22 GD32

HA11

5B065 BA01 CA30 EA05 EA12

5B083 AA05 BB03 CC01 CD06 EE08

EF14